**MERGE SORTED ARRAY**

You are given two integer arrays nums1 and nums2, sorted in **non-decreasing order**, and two integers m and n, representing the number of elements in nums1 and nums2 respectively.

**Merge** nums1 and nums2 into a single array sorted in **non-decreasing order**.

The final sorted array should not be returned by the function, but instead be *stored inside the array*nums1. To accommodate this, nums1 has a length of m + n, where the first m elements denote the elements that should be merged, and the last n elements are set to 0 and should be ignored. nums2 has a length of n.

**Example 1:**

**Input:** nums1 = [1,2,3,0,0,0], m = 3, nums2 = [2,5,6], n = 3

**Output:** [1,2,2,3,5,6]

**Explanation:** The arrays we are merging are [1,2,3] and [2,5,6].

The result of the merge is [1,2,2,3,5,6] with the underlined elements coming from nums1.

**Example 2:**

**Input:** nums1 = [1], m = 1, nums2 = [], n = 0

**Output:** [1]

**Explanation:** The arrays we are merging are [1] and [].

The result of the merge is [1].

**Example 3:**

**Input:** nums1 = [0], m = 0, nums2 = [1], n = 1

**Output:** [1]

**Explanation:** The arrays we are merging are [] and [1].

The result of the merge is [1].

Note that because m = 0, there are no elements in nums1. The 0 is only there to ensure the merge result can fit in nums1.

**CODE**

class Solution {

public:

    void merge(vector<int>& nums1, int m, vector<int>& nums2, int n)

    {

        int i,j=0,k=0,z=0;

        int n1[1000],n2[1000],n3[1000];

        for(i=0;i<m;i++)

        {

            n1[i]=nums1[i];

        }

        for(j=0;j<n;j++)

        {

            n2[j]=nums2[j];

        }

        i = 0;

        j = 0;

        while (i < m && j < n)

        {

            if (n1[i] <= n2[j])

            {

                n3[z++] = n1[i++];

            }

            else

            {

                n3[z++] = n2[j++];

            }

        }

        while (i < m || j < n)

        {

            if (i < m)

            {

                n3[z++] = n1[i++];

            }

            else

            {

                n3[z++] = n2[j++];

            }

        }

        for (i = 0; i < m + n; i++)

        {

            nums1[i] = n3[i];

        }

    }

};